Self-Supervised Learning and Pre-Training on Graphs

Liang Wang

Content

- Background
  - Self-Supervised Learning
  - Pre-Training
  - Relationship between SSL and PT
- Graph Self-Supervised Learning
  - Generative Learning based
  - Contrastive Learning based
- GNN Pre-Training
  - Homogeneous Graphs
    - Generative Learning based
    - Contrastive Learning based
  - Heterogeneous Graphs
    - only Contrastive Learning based

1. Background

Self-Supervised Learning

- **Self-supervised learning** (SSL) is a method of machine learning. It learns from unlabeled sample data. It can be regarded as an intermediate form between supervised and unsupervised learning. It is based on an artificial neural network.[1]

Self-Supervised Learning

	Self-Supervised Learning	Unsupervised Leaning
Similarities	• 数据标签：都不需要人工标注	
Differences	• 监督信号：SSL requires supervision unlike unsupervised learning • 学习目标：the objective of unsupervised learning is to identify the hidden patterns while the objective of SSL is to learn meaningful representations.	

	Self-Supervised Learning	Supervised Leaning
Similarities	• 监督信号：都有监督信号	
Differences	• 数据标签：In SSL, the labels are automatically generated based on data attributes and the definition of pretraining task without any human involvement. • 学习目标：the goal of supervised learning is provide task specific knowledge while SSL aims to provide the model with universal knowledge	

SSL does not require human labeled data and helps the model to gain more generalization ability (universal knowledge/ basic common sense/ background knowledge) by learning from large amounts of unlabeled data.

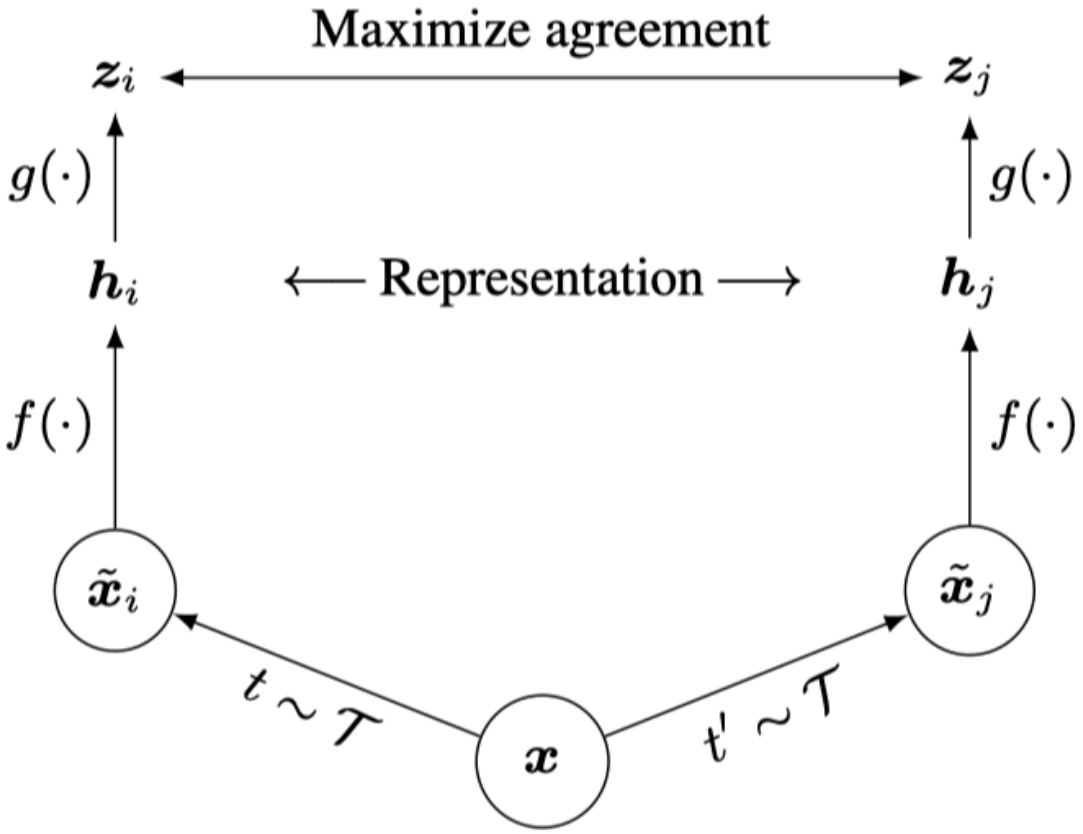
Self-Supervised Learning

- Types
  - Generative SSL
  - Contrastive SSL
  - Others: Adversarial SSL, Predictive SSL, …

Generative SSL

- 基本思想：基于重构，把编码后的数据解码/重构成输入数据
- 模型：Auto-Encoder, Variational Auto-Encoder
- 问题：重构误差小并不一定说明学习出来的特征好

Contrastive SSL

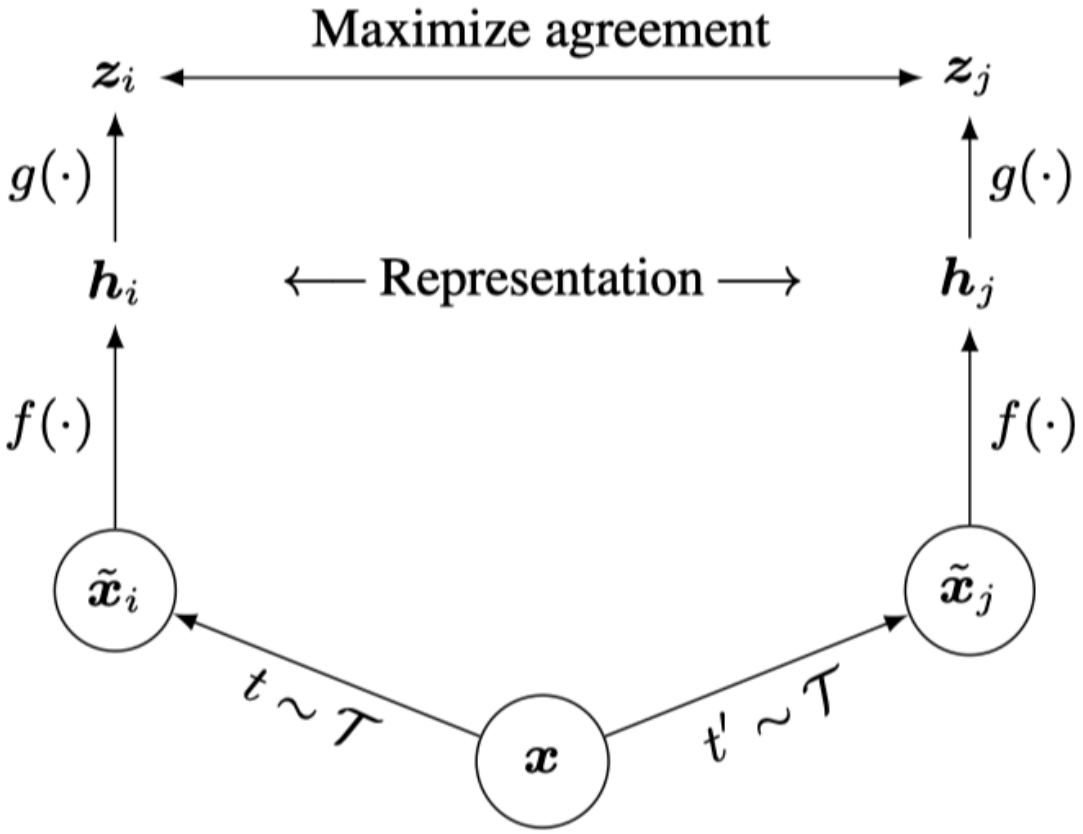


There are three main components:

- Data augmentation pipeline $\mathcal{T}$
- Encoder f and representation extractor g
- Contrastive mode and objective $\mathcal{L}$

A Simple Framework for Contrastive Learning of Visual Representations, ICML, 2020.

Contrastive SSL



Contrastive Objectives:

$$s(f(\boldsymbol{x}), f(\boldsymbol{x}^+)) \gg s(f(\boldsymbol{x}), f(\boldsymbol{x}^-))$$

InfoNCE Loss:

$$\mathcal{L} = -\mathbb{E}_X \left[\log \frac{\exp(s(\boldsymbol{x}, \boldsymbol{x}^+))}{\exp(s(\boldsymbol{x}, \boldsymbol{x}^+)) + \sum_{j=1}^{N-1} \exp(s(\boldsymbol{x}, \boldsymbol{x}_j))} \right]$$

A Simple Framework for Contrastive Learning of Visual Representations, ICML, 2020.

Pre-Training

- Learn universal language representations from large volumes of unlabeled text data and then transfer this knowledge to downstream tasks.
- Steps:
  - Prepare the pretraining corpus
  - Generate the vocabulary
  - Design the pretraining tasks (pretext tasks)
  - Choose the pretraining method
  - Adapt to downstream task

Ammus: A survey of transformer-based pretrained models in natural language processing, 2021.

Pre-Training

Design the pretraining tasks

- A pretraining task should be challenging enough so that it provides more training signals to the model. For example, tasks like MLM involves only 15% of tokens in each training sample for learning while tasks like Replaced Token Detection (RTD), Random Token Substitution (RTS), and Shuffled Token Detection (STD) involves all the tokens in the input sample for model learning.
- A pretraining task should be similar to the downstream task. For example, pretraining tasks like Seq2SeqLM or Denoising Auto Encoder (DAE) are similar to downstream tasks like text summarization, machine translation, etc.

Ammus: A survey of transformer-based pretrained models in natural language processing, 2021.

Pre-Training

Adapt to downstream task

- Feature-based

  generate contextual representations then used **as input features** in task-specific downstream models

- Fine-tuning

  Fine-tuning imparts task-specific knowledge to the model by **adapting its weights** based on task-specific loss.

- Prompt-based tuning

Ammus: A survey of transformer-based pretrained models in natural language processing, 2021.

Relationship between SSL and PT

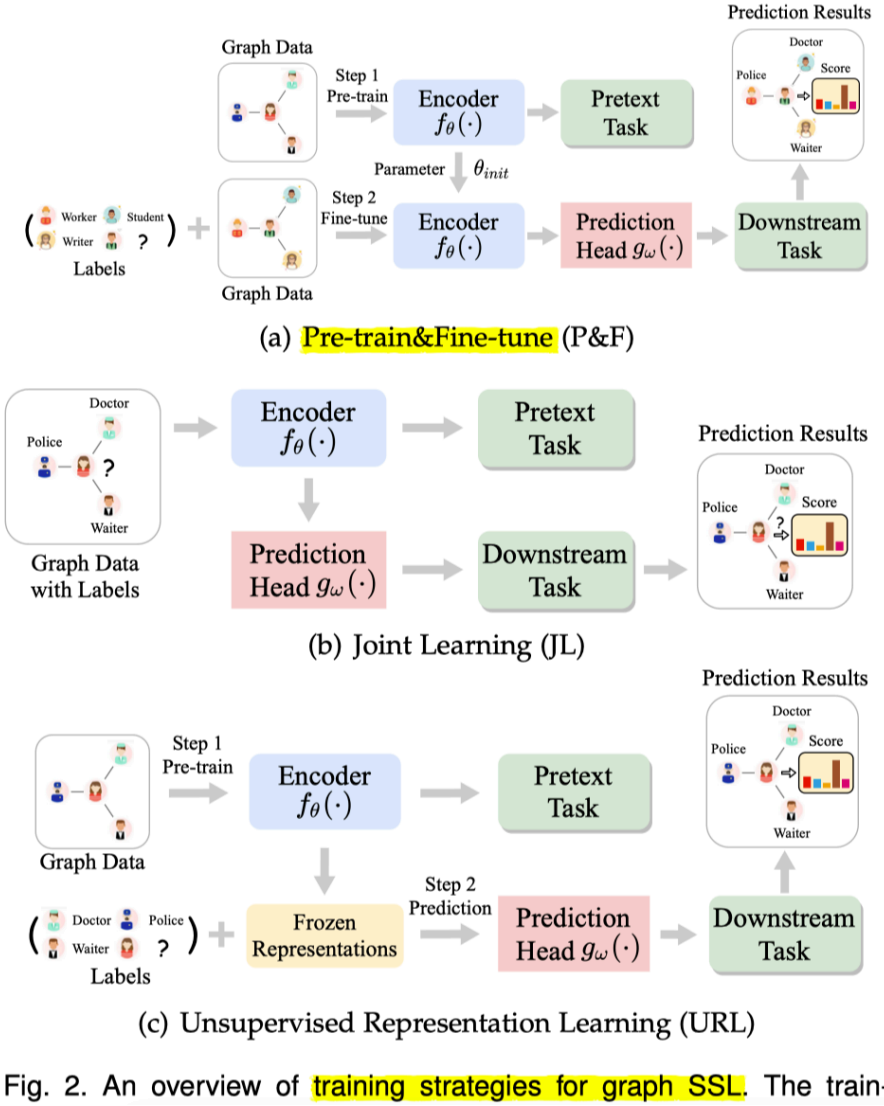
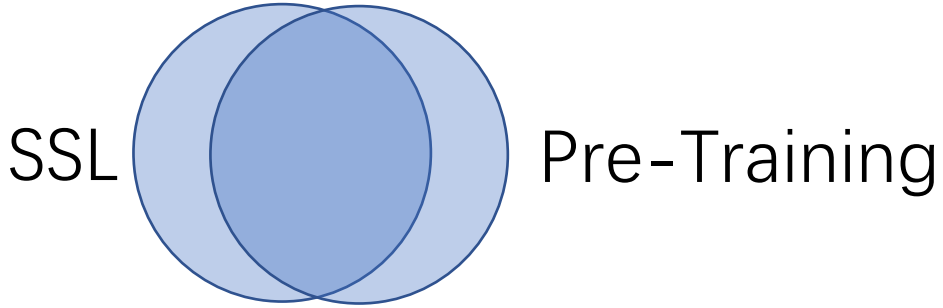


Fig. 2. An overview of training strategies for graph SSL. The train-



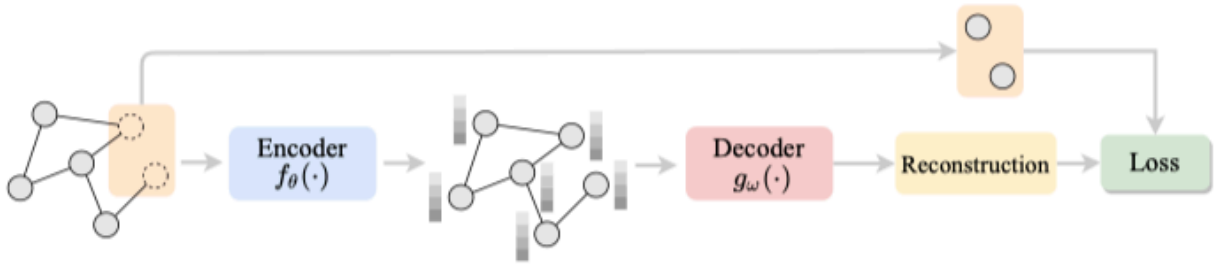
- From the perspective of SSL
  - Pre-training is one of the self-supervising learning training strategies
- From the perspective of Pre-Training
  - Self-supervised learning is one of the design of pre-training tasks and learning methods in pre-training

Self-supervised on Graphs: Contrastive, Generative, or Predictive, 2021.

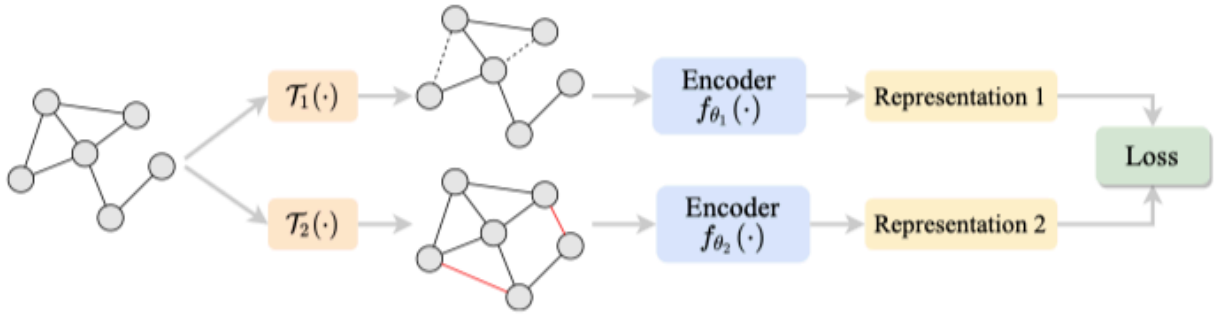
2. Graph Self-Supervised Learning

Graph Self-Supervised Learning

- Generative Learning based
  在原始空间度量损失
  - GAE：重构结构
  - GVAE：重构结构
  - GraphCNN：重构特征
- Contrastive Learning based
  在隐空间度量损失
  - DGI
  - InfoGraph
  - MVGRL
  - GCA
  - …



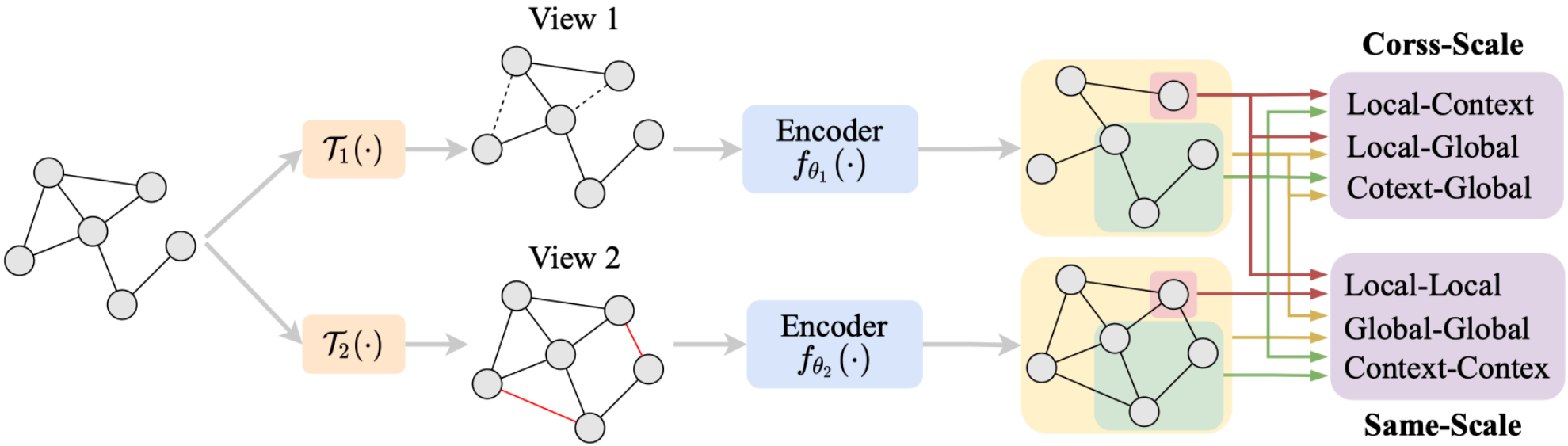
(b) Generative Method



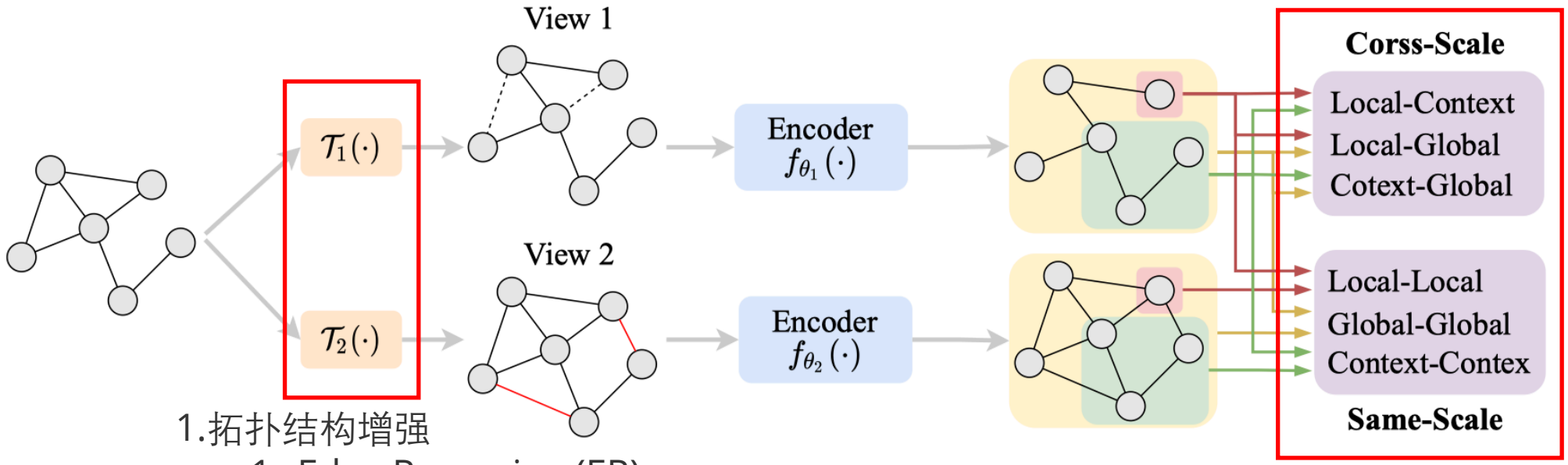
(a) Contrastive Method

Self-supervised on Graphs: Contrastive, Generative, or Predictive, 2021.

Graph Contrastive Self-Supervised Learning



Graph Contrastive Self-Supervised Learning



1.拓扑结构增强

1. Edge Removing (ER)
2. Edge Adding (EA)
3. Edge Flipping (EF)
4. Node Dropping (ND)
5. Subgraph induced by Random Walks (RWS)
6. diffusion with Personalized PageRank (PPR)
7. diffusion with Markov Diffusion Kernels (MDK)

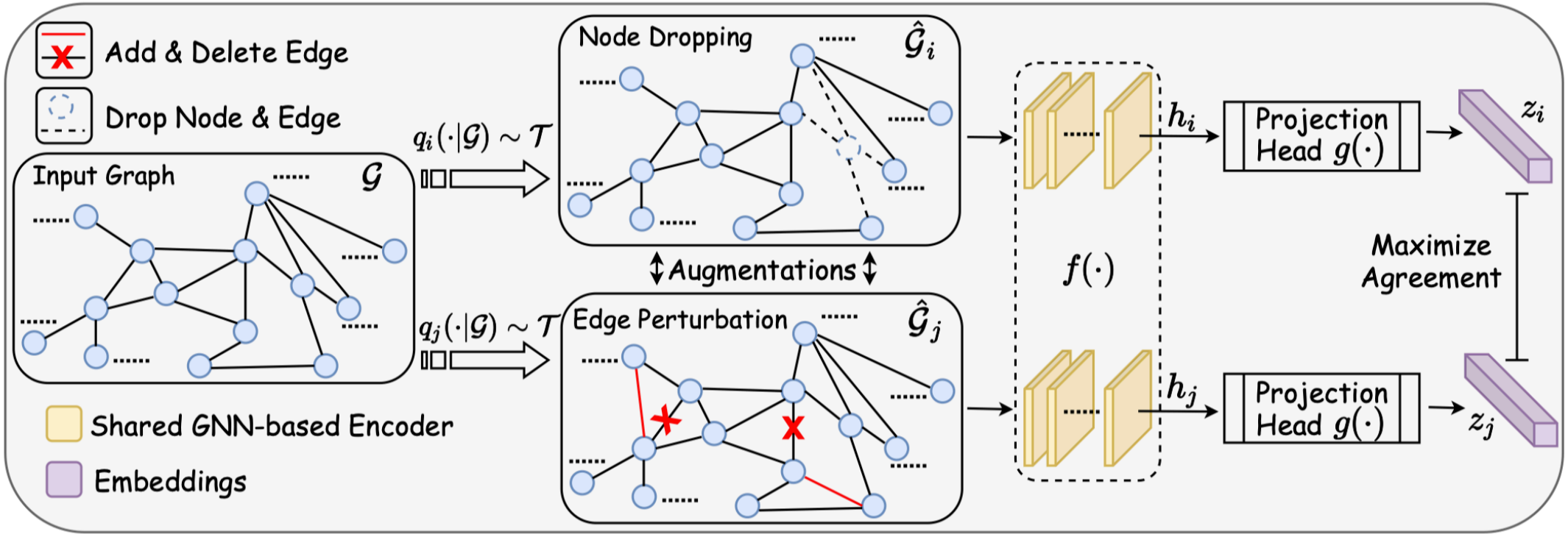
2.特征增强

1. Feature Masking (FM): 用0或高斯噪声mask
2. Feature Dropout (FD)

Local: 节点/边
Context: 子图
Global: 整图

Graph Contrastive Self-Supervised Learning

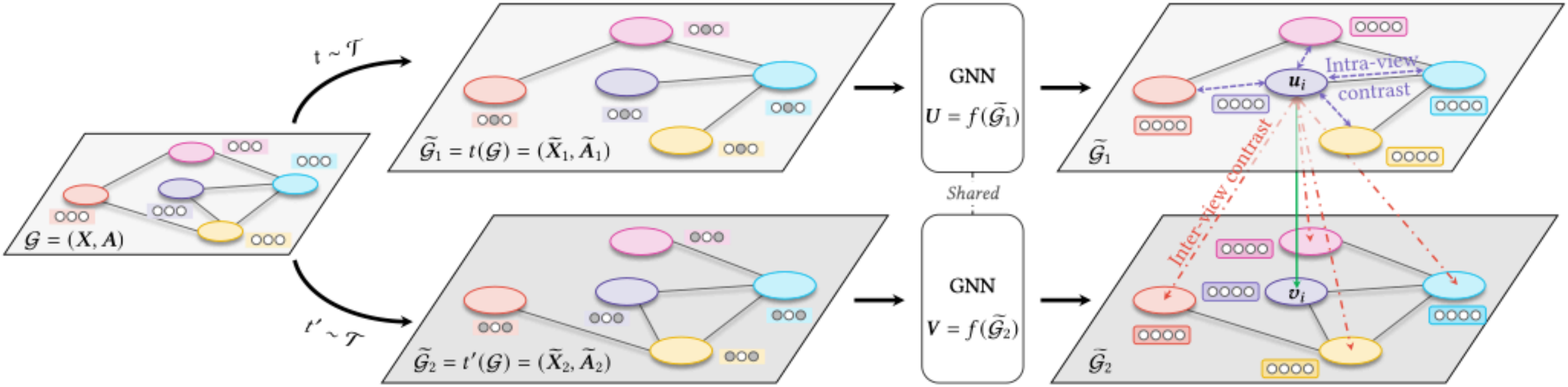
Graph Contrastive Learning with Augmentations, NIPS, 2020



GraphCL
Global-Global
Attribute Masking/Edge Pertubation/Random Walk Sampling

Graph Contrastive Self-Supervised Learning

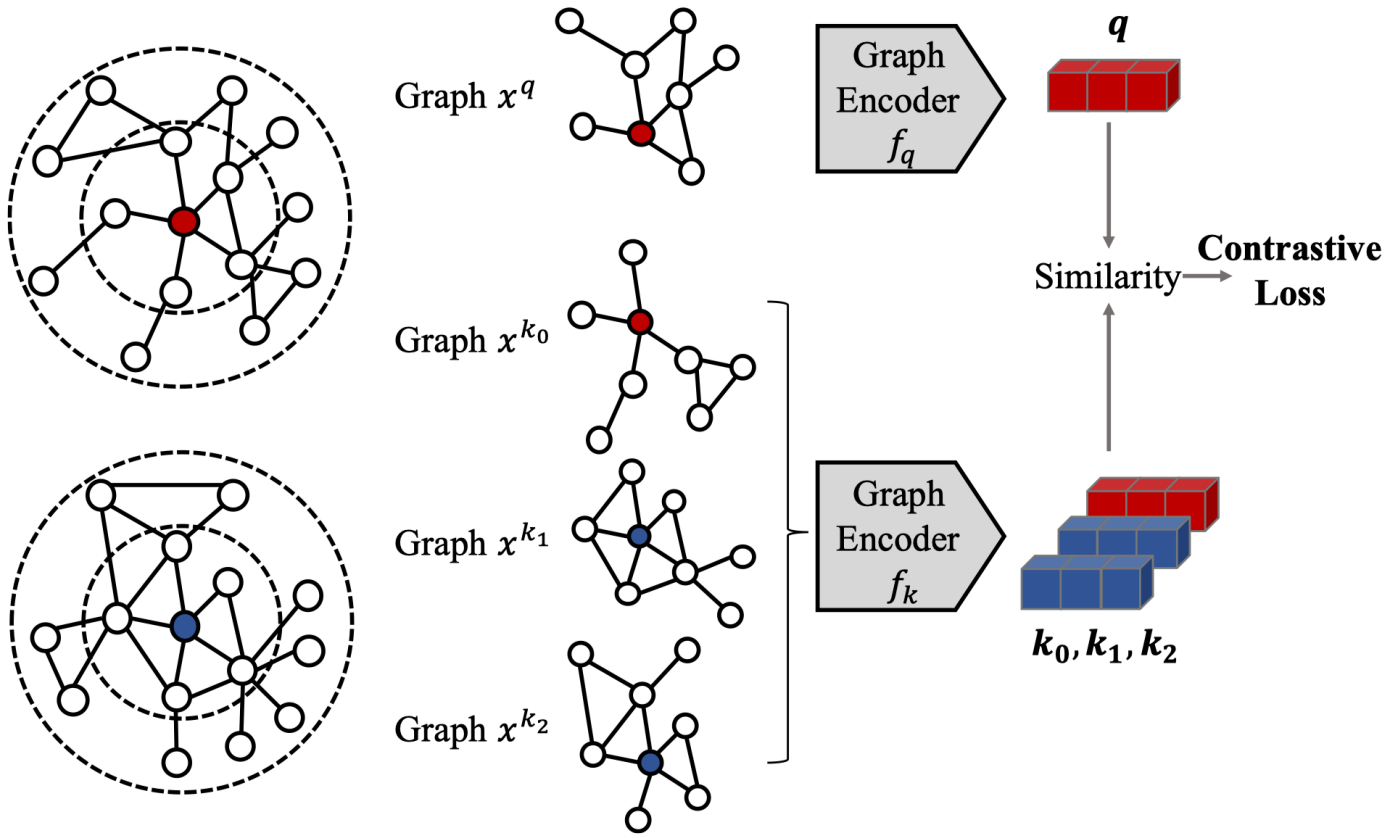
Graph contrastive learning with adaptive augmentation, WWW, 2021



GCA
Local-Local
Attention-based

Graph Contrastive Self-Supervised Learning

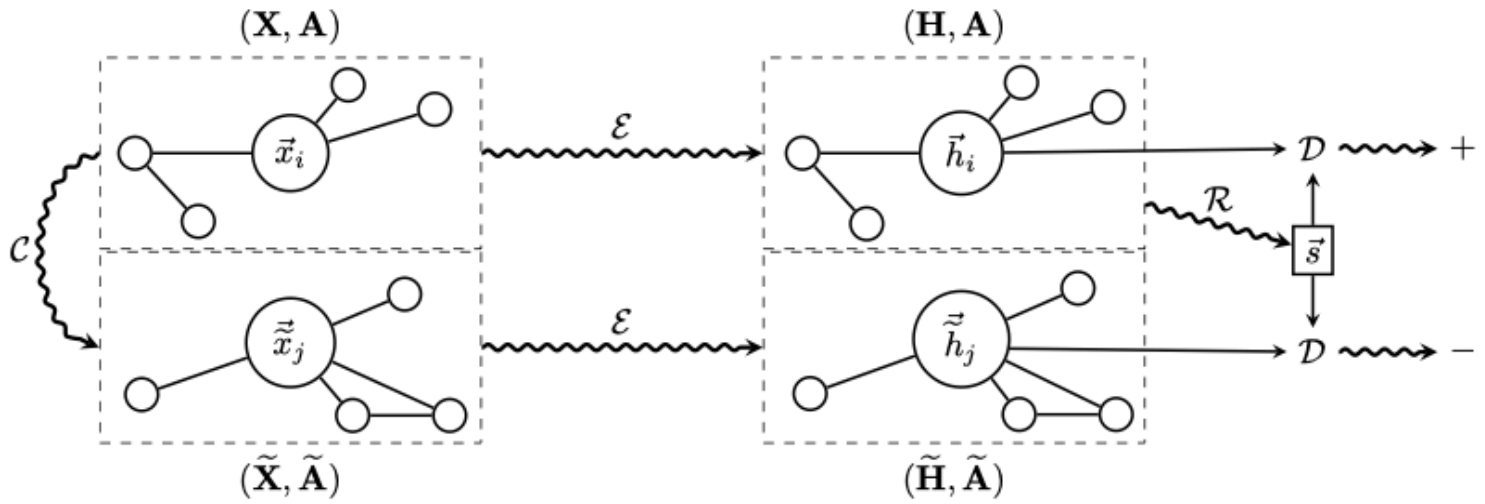
GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training, KDD, 2020



GCC
Context-Context
Random Walk Sampling

Graph Contrastive Self-Supervised Learning

Deep Graph Infomax, ICLR, 2019



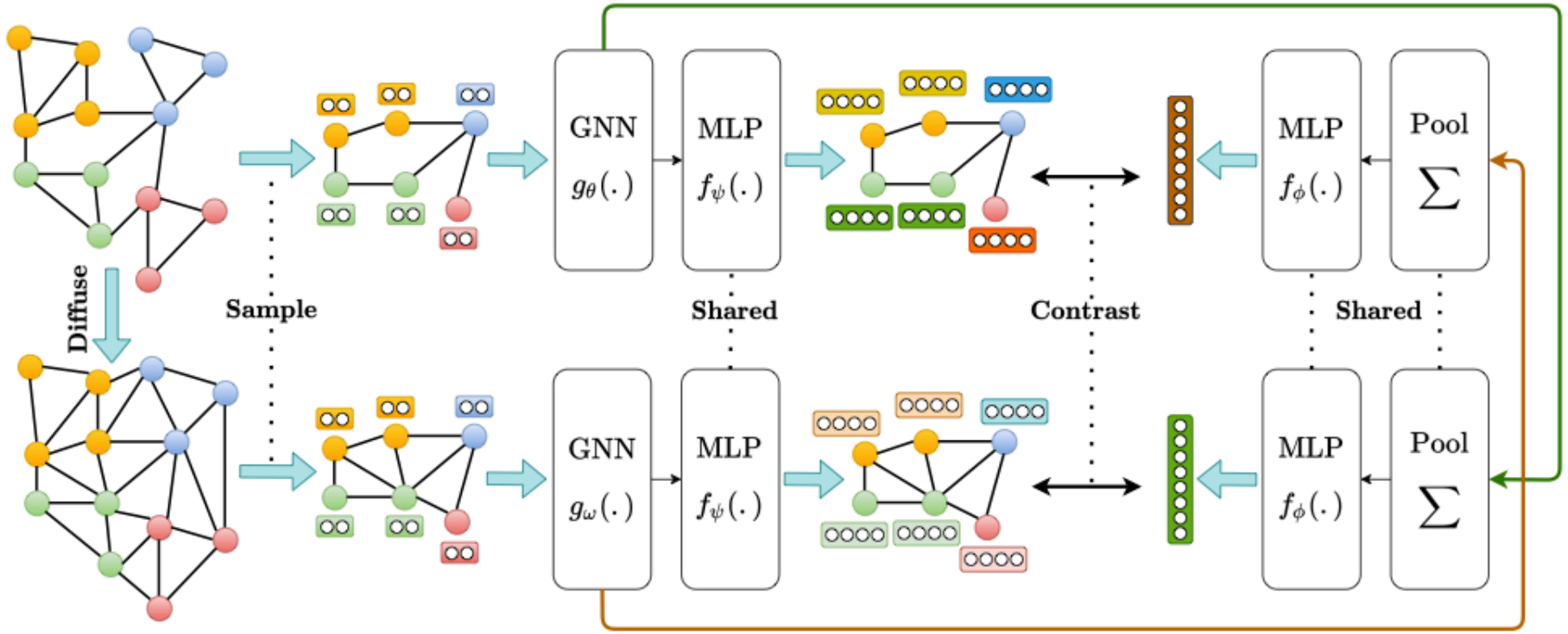
$$\mathcal{R}(\mathbf{H}) = \sigma\left(\frac{1}{N}\sum_{i=1}^{N}\vec{h}_i\right)$$

$$\mathcal{D}(\vec{h}_i, \vec{s}) = \sigma\left(\vec{h}_i^T \mathbf{W} \vec{s}\right)$$

DGI
Local-Global
Arbitrary

Graph Contrastive Self-Supervised Learning

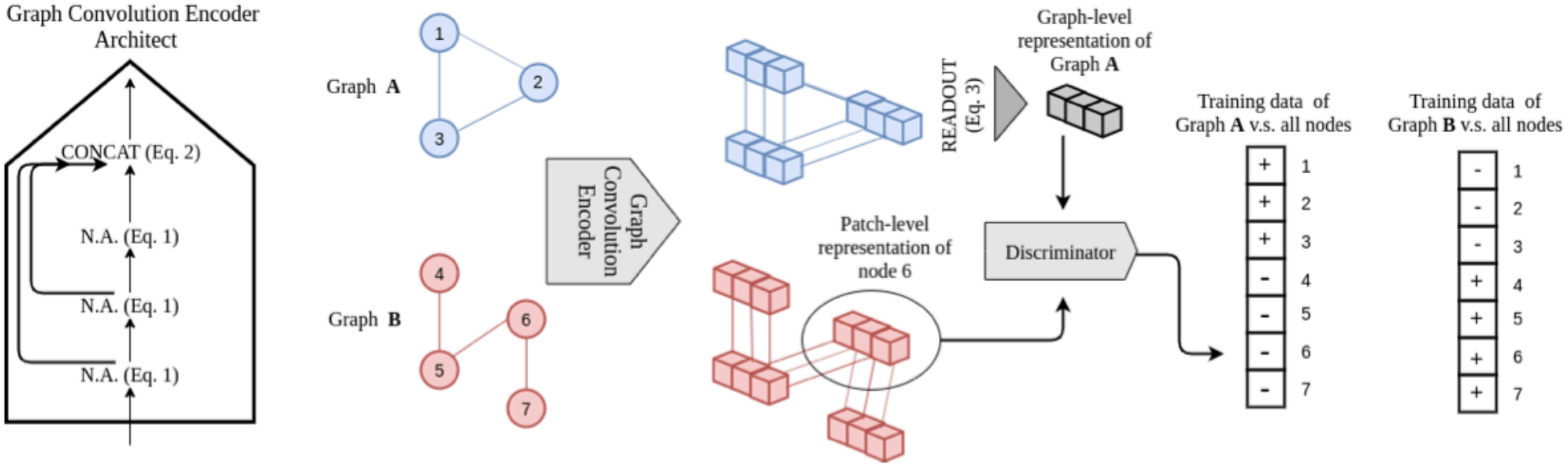
Contrastive Multi-View Representation Learning on Graphs, ICML, 2020



MVGRL
Local-Global
Attribute Masking/Edge Perturbating/Edge Diffusion/Random Walk Sampling

Graph Contrastive Self-Supervised Learning

InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization, ICLR, 2020



InfoGraph
Context-Global
None

3. GNN Pre-Training

GNN Pre-Training

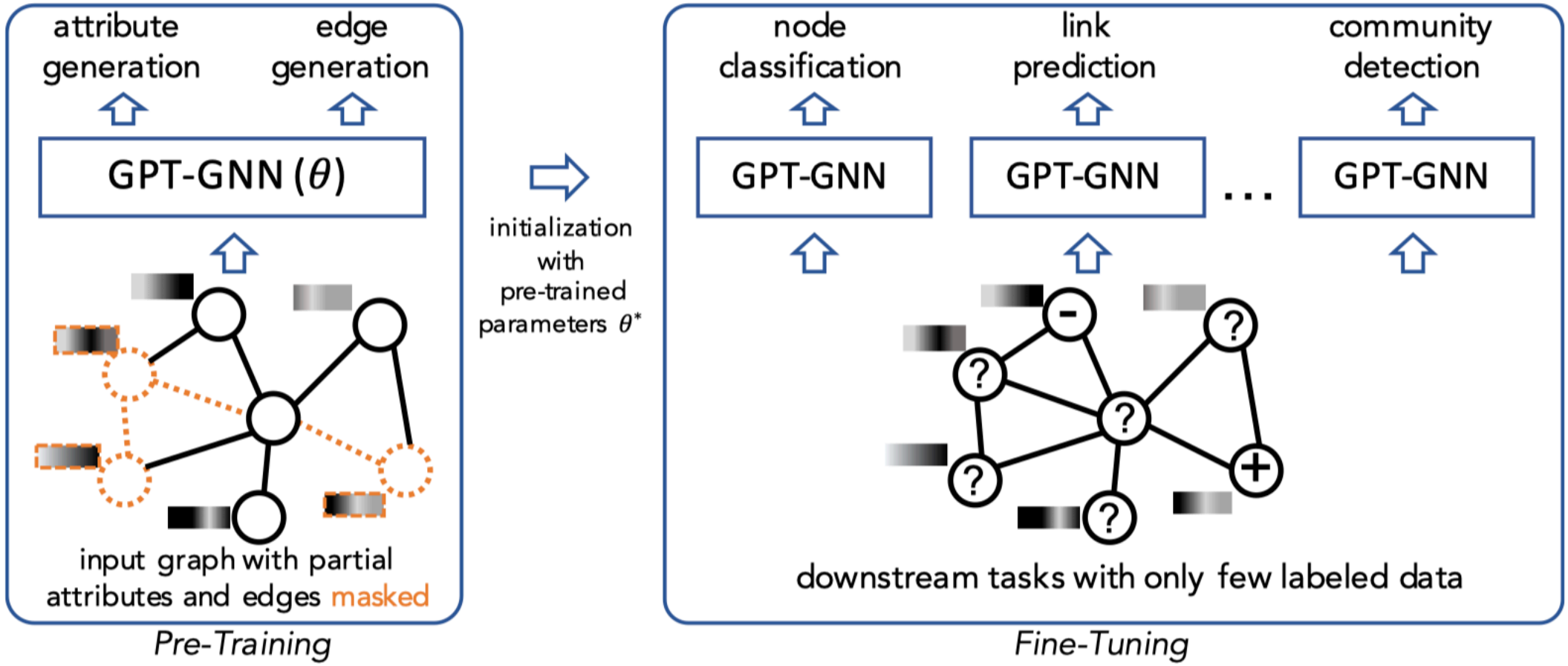
- pretext任务
  - 自监督范式：生成式/对比式
  - 学习的对象：attribute/structure
  - 学习的层次：local/context/global (node/edge/subgraph/graph)
- 适配下游任务的方式：feature-based/fine-tuning
- 图的类型：同质图/异质图

GNN Pre-Training

- Homogeneous Graphs
  - Generative Learning based
    - GPT-GNN, KDD, 2020
    - Graph Bert, 2020
  - Contrastive Learning based
    - Strategies for Pre-Training GNN, ICLR, 2020
- Heterogeneous Graphs
  - only Contrastive Learning based
    - PT-HGNN, KDD, 2021
    - CPT-HG, CIKM, 2021

GNN Pre-Training: 同质图+生成式方法

GPT-GNN: Generative Pre-Training of Graph Neural Networks, KDD, 2020



GNN Pre-Training: 同质图+生成式方法

GPT-GNN: Generative Pre-Training of Graph Neural Networks, KDD, 2020

图生成：$\theta^* = \max_\theta p(G;\theta)$

$$p(G;\theta) = \mathbb{E}_\pi \left[p_\theta(X^\pi, E^\pi) \right]$$

每轮迭代生成一个节点：$\log p_\theta(X,E) = \sum_{i=1}^{|\mathcal{V}|} \log p_\theta(X_i, E_i \mid X_{<i}, E_{<i})$

Naïve solution:

$$p_\theta(X_i, E_i | X_{<i}, E_{<i}) = p_\theta(X_i | X_{<i}, E_{<i}) \cdot p_\theta(E_i | X_{<i}, E_{<i})$$

忽视了结构和属性之间的关联

GNN Pre-Training: 同质图+生成式方法

GPT-GNN: Generative Pre-Training of Graph Neural Networks, KDD, 2020

图生成： $\theta^* = \max_\theta p(G;\theta)$

$$p(G;\theta) = \mathbb{E}_\pi \left[p_\theta(X^\pi, E^\pi) \right]$$

每轮迭代生成一个节点： $$\log p_\theta(X,E) = \sum_{i=1}^{|\mathcal{V}|} \log p_\theta(X_i, E_i \mid X_{<i}, E_{<i})$$

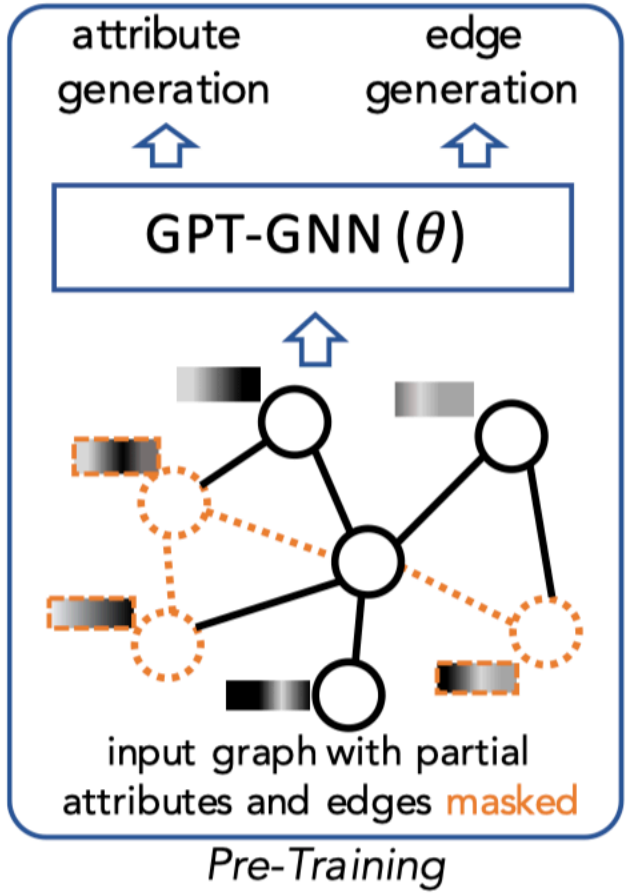
本文提出 dependency-aware factorization mechanism for the attributed graph generation，把生成任务分解为两部分：

- given the observed edges, generate node attributes;
- given the observed edges and generated node attributes, generate the remaining edges.

$$\begin{aligned}
&p_\theta(X_i, E_i \mid X_{<i}, E_{<i}) \\
=&\sum\nolimits_o p_\theta(X_i, E_{i,\neg o} \mid E_{i,o}, X_{<i}, E_{<i}) \cdot p_\theta(E_{i,o} \mid X_{<i}, E_{<i}) \\
=&\mathbb{E}_o \left[p_\theta(X_i, E_{i,\neg o} \mid E_{i,o}, X_{<i}, E_{<i}) \right] \\
=&\mathbb{E}_o \Big[\underbrace{p_\theta(X_i \mid E_{i,o}, X_{<i}, E_{<i})}_{\text{1) generate attributes}} \cdot \underbrace{p_\theta(E_{i,\neg o} \mid E_{i,o}, X_{\leq i}, E_{<i})}_{\text{2) generate edges}} \Big].
\end{aligned}$$

GNN Pre-Training: 同质图+生成式方法

GPT-GNN: Generative Pre-Training of Graph Neural Networks, KDD, 2020



Efficient Attribute and Edge Generation

目标：compute the loss of attribute and edge generations by running the GNN only once for the input graph

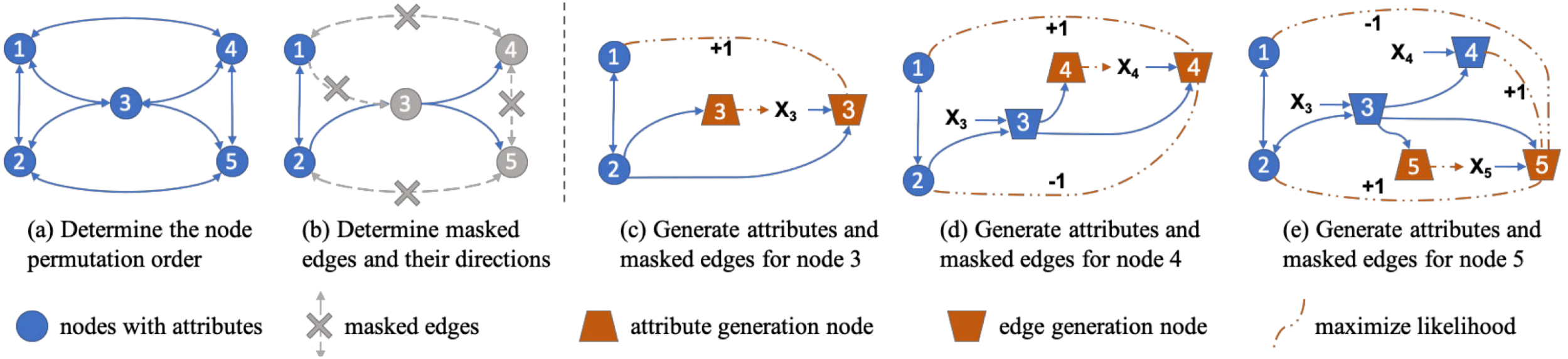
问题： we expect to conduct attribute generation and edge generation simultaneously. However, edge generation requires node attributes as input, which can be leaked to attribute generation.

方法：
separate each node into two types:

- Attribute Generation Nodes. （特征随机初始化）
- Edge Generation Nodes. （采用真实特征）

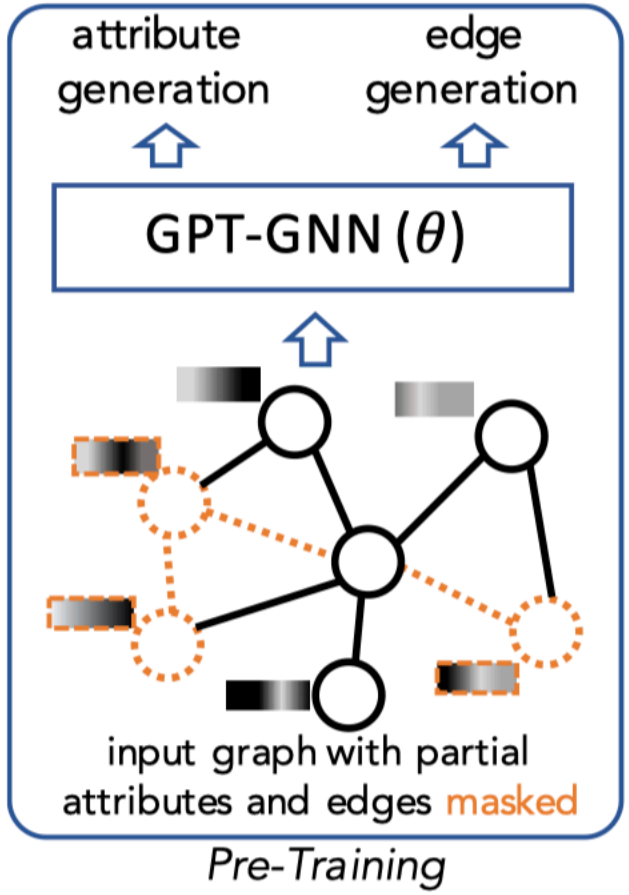
GNN Pre-Training: 同质图+生成式方法

GPT-GNN: Generative Pre-Training of Graph Neural Networks, KDD, 2020



GNN Pre-Training: 同质图+生成式方法

GPT-GNN: Generative Pre-Training of Graph Neural Networks, KDD, 2020



属性重构损失（相当于期望中的第一项）：

$$\mathcal{L}_i^{Attr} = Distance\left(Dec^{Attr}(h_i^{Attr}), X_i\right)$$

边结构重构：
假设每个边的重构是独立的，所以期望中第二项分解为：

$$p_\theta(E_{i,\neg o} \mid E_{i,o}, X_{\leq i}, E_{<i}) = \prod_{j^+ \in E_{i,\neg o}} p_\theta(j^+ \mid E_{i,o}, X_{\leq i}, E_{<i}).$$

重构损失：

$$\mathcal{L}_i^{Edge} = -\sum_{j^+ \in E_{i,\neg o}} \log \frac{\exp\left(Dec^{Edge}(h_i^{Edge}, h_{j^+}^{Edge})\right)}{\sum_{j \in S_i^- \cup \{j^+\}} \exp\left(Dec^{Edge}(h_i^{Edge}, h_j^{Edge})\right)}$$

GNN Pre-Training: 同质图+生成式方法

GPT-GNN: Generative Pre-Training of Graph Neural Networks, KDD, 2020

	Downstream Dataset	OAG			Amazon		
	Evaluation Task	Paper–Field	Paper–Venue	Author ND	Fashion	Beauty	Luxury
	No Pre-train	.336±.149	.365±.122	.794±.105	.586±.074	.546±.071	.494±.067
Field Transfer	GAE	.403±.114	.418±.093	.816±.084	.610±.070	.568±.066	.516±.071
	GraphSAGE (unsp.)	.368±.125	.401±.096	.803±.092	.597±.065	.554±.061	.509±.052
	Graph Infomax	.387±.112	.404±.097	.810±.084	.604±.063	.561±.063	.506±.074
	GPT-GNN (Attr)	.396±.118	.423±.105	.818±.086	.621±.053	.576±.056	.528±.061
	GPT-GNN (Edge)	.401±.109	.428±.096	.826±.093	.616±.060	.570±.059	.520±.047
	GPT-GNN	**.407±.107**	**.432±.098**	**.831±.102**	**.625±.055**	**.577±.054**	**.531±.043**
Time Transfer	GAE	.384±.117	.412±.101	.812±.095	.603±.065	.562±.063	.510±.071
	GraphSAGE (unsp.)	.352±.121	.394±.105	.799±.093	.594±.067	.553±.069	.501±.064
	Graph Infomax	.369±.116	.398±.102	.805±.089	.599±.063	.558±.060	.503±.063
	GPT-GNN (Attr)	.382±.114	.414±.098	.811±.089	.614±.057	**.573±.053**	.522±.051
	GPT-GNN (Edge)	.392±.105	.421±.102	.821±.088	.608±.055	.567±.038	.513±.058
	GPT-GNN	**.400±.108**	**.429±.101**	**.825±.093**	**.617±.059**	.572±.059	**.525±.057**
Time + Field Transfer	GAE	.371±.124	.403±.108	.806±.102	.596±.065	.554±.063	.505±.061
	GraphSAGE (unsp.)	.349±.130	.393±.118	.797±.097	.589±.071	.545±.068	.498±.064
	Graph Infomax	.360±.121	.391±.102	.800±.093	.591±.068	.550±.058	.501±.063
	GPT-GNN (Attr)	.364±.115	.409±.103	.809±.094	.608±.062	.569±.057	.517±.057
	— (w/o node separation)	.347±.128	.391±.102	.791±.108	.585±.068	.546±.062	.497±.062
	GPT-GNN (Edge)	.386±.116	.414±.104	.815±.105	.604±.058	.565±.057	.514±.047
	— (w/o adaptive queue)	.376±.121	.410±.115	.808±.104	.599±.068	.562±.065	.509±.062
	GPT-GNN	**.393±.112**	**.420±.108**	**.818±.102**	**.610±.054**	**.572±.063**	**.521±.049**

Table 1: Performance of different downstream tasks on OAG and Amazon by using different pre-training frameworks with the heterogeneous graph transformer (HGT) [15] as the base model. 10% of labeled data is used for fine-tuning.

GNN Pre-Training: 同质图+生成式方法

GPT-GNN: Generative Pre-Training of Graph Neural Networks, KDD, 2020

Algorithm 1 The GPT-GNN Pre-Training Framework

Require: Input Attributed Graph G, Graph Sampler $Sampler(\cdot)$.
Ensure:

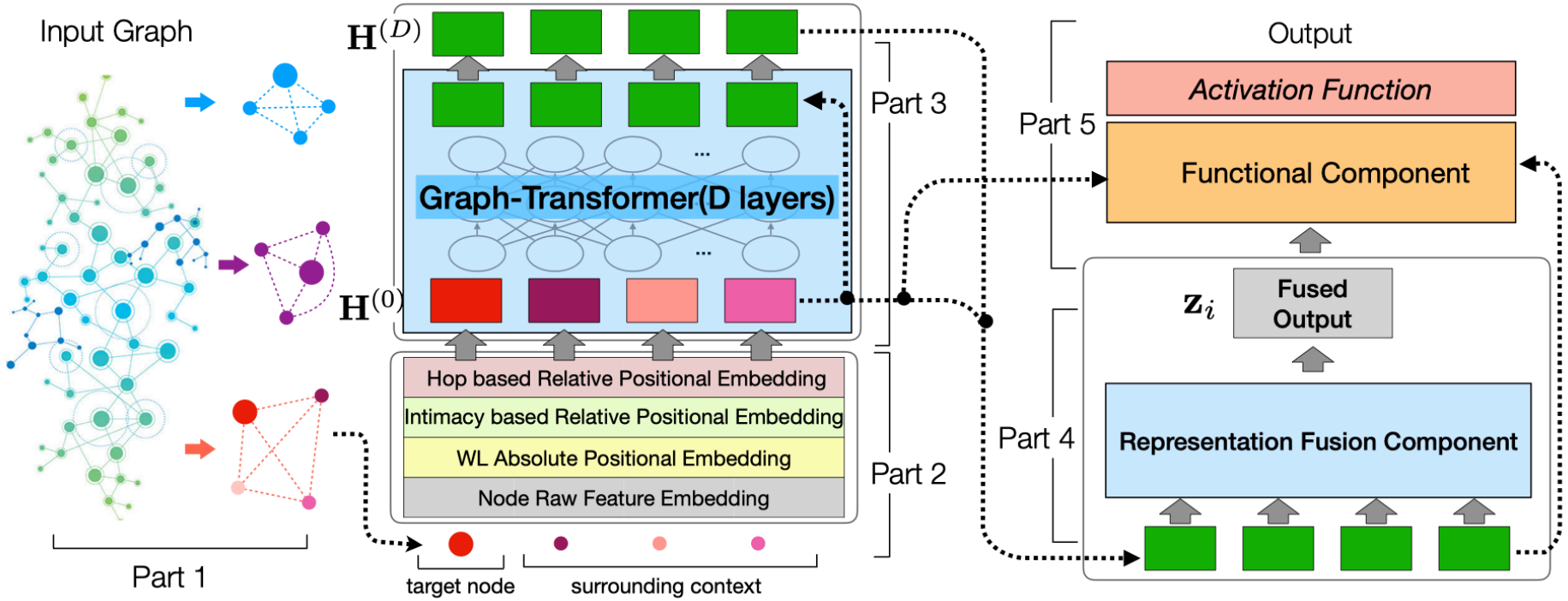
Initialize the GNN model as f_θ, the attribute generation decoder as Dec^{Attr}, and the edge generation decoder as Dec^{Edge}.
Initialize the adaptive node embedding queue $Q = \{\}$ and the attribute vector h^{init}.
for each sampled graph $\hat{G} \in Sampler(G)$ **do**
  For each node, sample the observed edge index o and masked edges $\neg o$, and delete masked edges $E_{i,\neg o}$ accordingly.
  Separate each node into the Attribute Generation and Edge Generation nodes. Replace the input to Attribute Generation node as h_{init}. Apply GNN f_θ to get two sets of node embeddings h^{Attr} and h^{Edge} for each node in the graph.
  for node i with attributes X_i and masked edges $E_{i,\neg o}$ **do**
    Calculate the attribute generation loss $\mathcal{L}^{Attr}$ by Eq. 4
    Prepare negative samples S_i^- for edge generation by concatenating unconnected nodes and adaptive queue Q.
    Calculate the edge generation loss $\mathcal{L}^{Edge}$ by Eq. 6
  end for
  Optimize θ by minimizing $\mathcal{L}^{Attr}$ and $\mathcal{L}^{Edge}$.
  Update Q by adding in h^{Edge} and popping out most outdated embeddings.
end for
return Pre-trained model parameters θ^* for downstream tasks

Adapt to downstream task: Fine-tuning

Base model: HGT

GNN Pre-Training: 同质图+生成式方法

Graph-Bert: Only Attention is Needed for Learning Graph Representations, KDD, 2020



(1) linkless subgraph batching, (2) node input embedding, (3) graph-transformer based encoder, (4) representation fusion, and (5) the functional component

GNN Pre-Training: 同质图+生成式方法

Graph-Bert: Only Attention is Needed for Learning Graph Representations, KDD, 2020

图分解成 linkless subgraph batches, 依赖 top-k intimacy sampling approach

$$\mathbf{S} = \alpha \cdot \left(\mathbf{I} - (1-\alpha) \cdot \bar{\mathbf{A}}\right)^{-1}$$

DEFINITION 1. *(Node Context): Given an input graph G and its intimacy matrix $\mathbf{S}$, for node v_i in the graph, we define its learning context as set $\Gamma(v_i) = \{v_j | v_j \in \mathcal{V} \setminus \{v_i\} \wedge \mathbf{S}(i,j) \geq \theta_i\}$. Here, the term θ_i defines the minimum intimacy score threshold for nodes to involve in v_i's context.*

of v_i in graph G. Based on the node context concept, we can also represent the set of sampled graph batches for all the nodes as set $\mathcal{G} = \{g_1, g_2, \cdots, g_{|\mathcal{V}|}\}$, and g_i denotes the subgraph sampled for v_i (as the target node). Formally, g_i can be represented as $g_i = (\mathcal{V}_i, \varnothing)$, where the node set $\mathcal{V}_i = \{v_i\} \cup \Gamma(v_i)$ covers both v_i and its context nodes and the link set is null.

GNN Pre-Training: 同质图+生成式方法

Graph-Bert: Only Attention is Needed for Learning Graph Representations, KDD, 2020

Node Input Vector Embeddings
(1) raw feature vector embedding,
(2) Weisfeiler-Lehman absolute role embedding
(3) intimacy based relative positional embedding
(4) hop based relative distance embedding

GNN Pre-Training: 同质图+生成式方法

Graph-Bert: Only Attention is Needed for Learning Graph Representations, KDD, 2020

Graph Transformer based Encoder

$$\begin{cases} \mathbf{H}^{(0)} = [\mathbf{h}_i^{(0)}, \mathbf{h}_{i,1}^{(0)}, \cdots, \mathbf{h}_{i,k}^{(0)}]^\top, \\ \mathbf{H}^{(l)} = \text{G-Transformer}\left(\mathbf{H}^{(l-1)}\right), \forall l \in \{1, 2, \cdots, D\} \\ \mathbf{z}_i \quad = \text{Fusion}\left(\mathbf{H}^{(D)}\right). \end{cases}$$

$$\mathbf{h}_j^{(0)} = \text{Aggregate}(\mathbf{e}_j^{(x)}, \mathbf{e}_j^{(r)}, \mathbf{e}_j^{(p)}, \mathbf{e}_j^{(d)}).$$

$$\begin{aligned} \mathbf{H}^{(l)} &= \text{G-Transformer}\left(\mathbf{H}^{(l-1)}\right) \\ &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}}\right)\mathbf{V} + \text{G-Res}\left(\mathbf{H}^{(l-1)}, \mathbf{X}_i\right), \end{aligned}$$

where

$$\begin{cases} \mathbf{Q} &= \mathbf{H}^{(l-1)}\mathbf{W}_Q^{(l)}, \\ \mathbf{K} &= \mathbf{H}^{(l-1)}\mathbf{W}_K^{(l)}, \\ \mathbf{V} &= \mathbf{H}^{(l-1)}\mathbf{W}_V^{(l)}. \end{cases}$$

GNN Pre-Training: 同质图+生成式方法

Graph-Bert: Only Attention is Needed for Learning Graph Representations, KDD, 2020

Pre-Trained Tasks:
(1) node attribute reconstruction
(2) graph structure recovery

$$\hat{\mathbf{x}}_i = \mathrm{FC}(\mathbf{z}_i)$$

$$\ell_1 = \frac{1}{|\mathcal{V}|} \sum_{v_i \in \mathcal{V}} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2 .$$

$$\hat{s}_{i,j} = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\| \|\mathbf{z}_j\|}$$

$$\ell_2 = \frac{1}{|\mathcal{V}|^2} \left\|\mathbf{S} - \hat{\mathbf{S}}\right\|_F^2 ,$$

Adapt to downstream task: Feature-based

4.2 Model Transfer and Fine-tuning

In applying the learned GRAPH-BERT into new learning tasks, the learned graph representations can be either fed into the new tasks directly or with necessary adjustment, i.e., fine-tuning. In this part, we can take the *node classification* and *graph clustering* tasks as the examples, where *graph clustering* can use the learned representations directly but fine-tuning will be necessary for the *node classification* task.

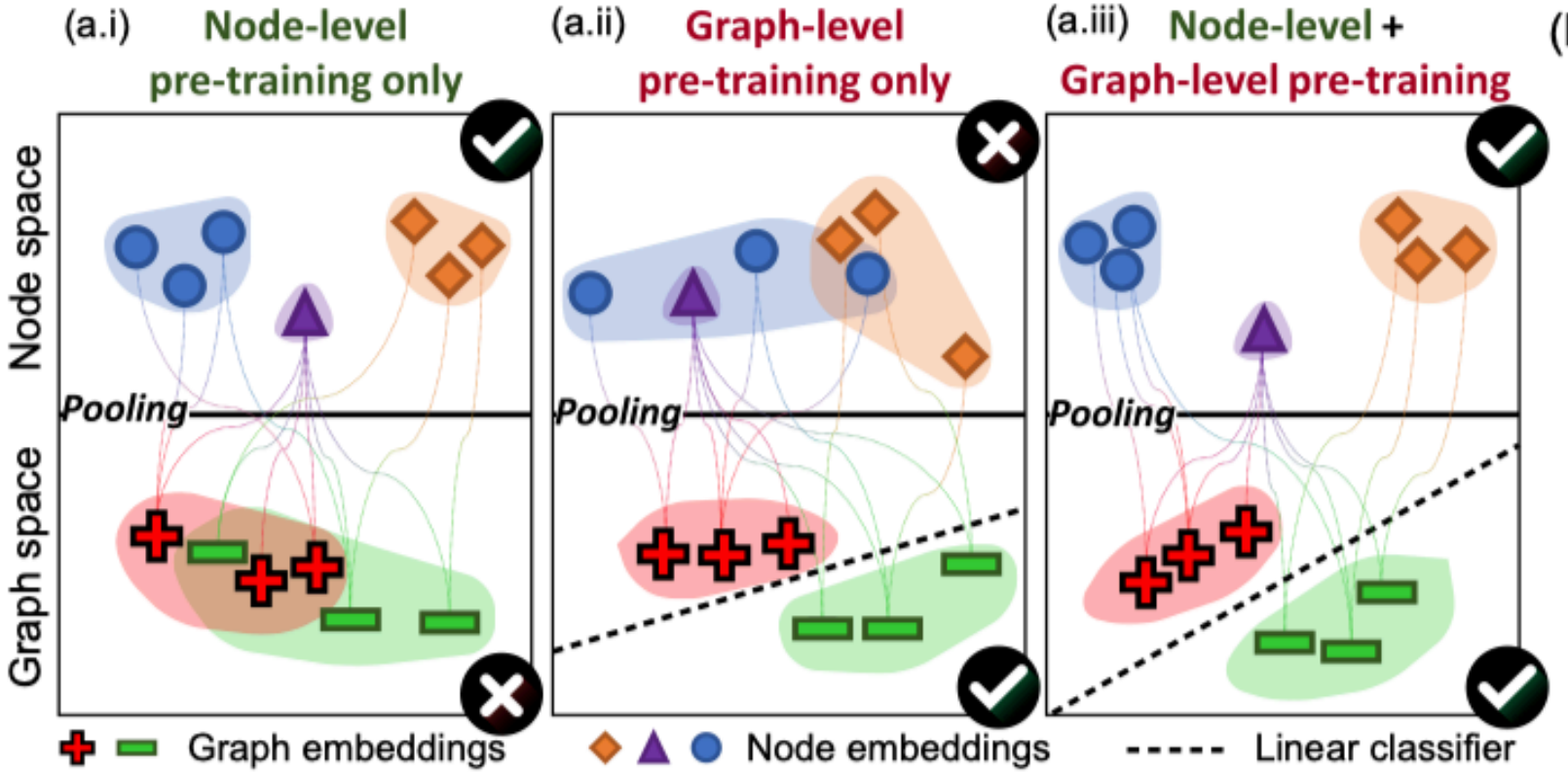
GNN Pre-Training: 同质图+生成式方法

Graph-Bert: Only Attention is Needed for Learning Graph Representations, KDD, 2020

Methods		Datasets (Accuracy/Rand & Epoch)					
Pre-Train Task	Fine-Tune Task	**Cora**		**Citeseer**		**Pubmed**	
Node Reconstruction	Node Classification	0.827	30	0.649	400	0.780	100
	Graph Clustering	**0.400**	30	**0.312**	400	0.027	100
Structural Recovery	Node Classification	0.823	30	0.662	400	0.788	100
	Graph Clustering	0.123	30	0.090	400	0.132	100
Both	Node Classification	**0.836**	30	**0.672**	400	**0.791**	100
	Graph Clustering	0.177	30	0.203	400	0.159	100
None	Node Classification	0.805	30	0.654	400	0.786	100
	Graph Clustering	0.080	30	0.249	400	**0.281**	100

GNN Pre-Training: 同质图+对比式方法

Strategies for Pre-Training Graph Neural Networks, ICLR, 2020

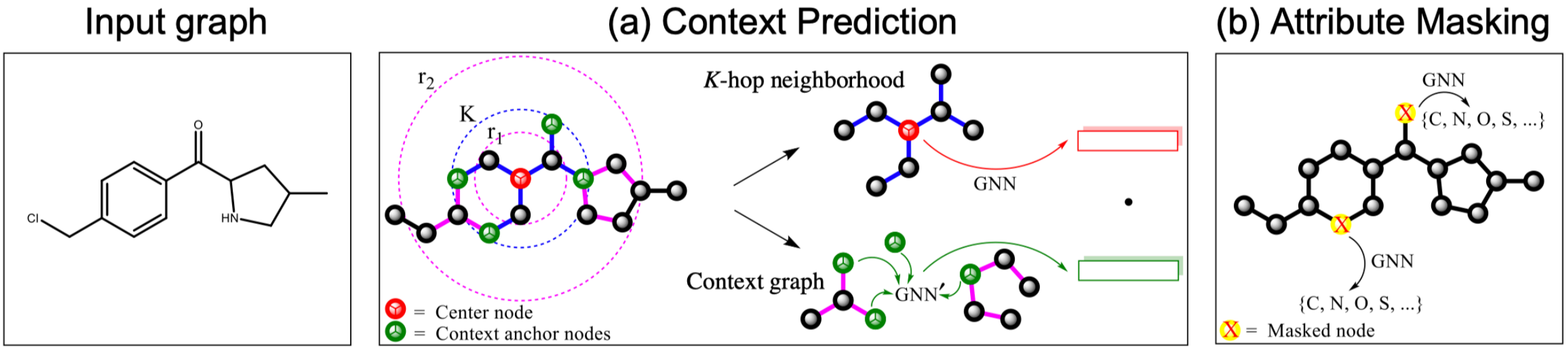


(b) Categorization of our pre-training methods

	Node-level	Graph-level
Attribute prediction	Attribute Masking	Supervised Attribute Prediction
Structure prediction	Context Prediction	Structural Similarity Prediction

GNN Pre-Training: 同质图+对比式方法

Strategies for Pre-Training Graph Neural Networks, ICLR, 2020



$$\sigma\left(h_v^{(K)\top} c_{v'}^{G'}\right) \approx \mathbf{1}\{v \text{ and } v' \text{ are the same nodes}\},$$

node-level预训练

上下文预测任务(Local-Context Contrast)：使处于相似上下文结构的节点被图神经网络映射到相近的空间

特征预测任务(Generate)：使图神经网络模型能够通过局部结构预测点或者边的特征

graph-level预训练

一系列图的有监督任务。例如，训练图神经网络模型预测化学分子的已知特性或者蛋白质的已知功能。

GNN Pre-Training: 同质图+对比式方法

Strategies for Pre-Training Graph Neural Networks, ICLR, 2020

Dataset		BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE	**Average**
# Molecules		2039	7831	8575	1427	1478	93087	41127	1513	/
# Binary prediction tasks		1	12	617	27	2	17	1	1	/
Pre-training strategy		Out-of-distribution prediction (scaffold split)								
Graph-level	Node-level									
–	–	65.8 ±4.5	74.0 ±0.8	63.4 ±0.6	57.3 ±1.6	58.0 ±4.4	71.8 ±2.5	75.3 ±1.9	70.1 ±5.4	67.0
–	Infomax	**68.8 ±0.8**	75.3 ±0.5	62.7 ±0.4	58.4 ±0.8	69.9 ±3.0	75.3 ±2.5	76.0 ±0.7	75.9 ±1.6	70.3
–	EdgePred	67.3 ±2.4	76.0 ±0.6	64.1 ±0.6	60.4 ±0.7	64.1 ±3.7	74.1 ±2.1	76.3 ±1.0	79.9 ±0.9	70.3
–	AttrMasking	64.3 ±2.8	76.7 ±0.4	64.2 ±0.5	61.0 ±0.7	71.8 ±4.1	74.7 ±1.4	77.2 ±1.1	79.3 ±1.6	71.1
–	ContextPred	68.0 ±2.0	75.7 ±0.7	63.9 ±0.6	60.9 ±0.6	65.9 ±3.8	75.8 ±1.7	77.3 ±1.0	79.6 ±1.2	70.9
Supervised	–	68.3 ±0.7	77.0 ±0.3	64.4 ±0.4	62.1 ±0.5	57.2 ±2.5	79.4 ±1.3	74.4 ±1.2	76.9 ±1.0	70.0
Supervised	Infomax	68.0 ±1.8	77.8 ±0.3	64.9 ±0.7	60.9 ±0.6	**71.2 ±2.8**	**81.3 ±1.4**	77.8 ±0.9	80.1 ±0.9	72.8
Supervised	EdgePred	66.6 ±2.2	**78.3 ±0.3**	**66.5 ±0.3**	**63.3 ±0.9**	70.9 ±4.6	78.5 ±2.4	77.5 ±0.8	79.1 ±3.7	72.6
Supervised	AttrMasking	66.5 ±2.5	77.9 ±0.4	65.1 ±0.3	**63.9 ±0.9**	**73.7 ±2.8**	**81.2 ±1.9**	77.1 ±1.2	80.3 ±0.9	73.2
Supervised	ContextPred	**68.7 ±1.3**	**78.1 ±0.6**	65.7 ±0.6	62.7 ±0.8	**72.6 ±1.5**	**81.3 ±2.1**	**79.9 ±0.7**	**84.5 ±0.7**	**74.2**

Table 1: **Test ROC-AUC (%) performance on molecular prediction benchmarks using different pre-training strategies with GIN.** The rightmost column averages the mean of test performance

GNN Pre-Training: 同质图+对比式方法

Strategies for Pre-Training Graph Neural Networks, ICLR, 2020

	Chemistry			Biology		
	Non-pre-trained	Pre-trained	Gain	Non-pre-trained	Pre-trained	Gain
GIN	67.0	**74.2**	**+7.2**	64.8 ± 1.0	**74.2 ± 1.5**	**+9.4**
GCN	**68.9**	72.2	+3.4	63.2 ± 1.0	70.9 ± 1.7	+7.7
GraphSAGE	68.3	70.3	+2.0	65.7 ± 1.2	68.5 ± 1.5	+2.8
GAT	66.8	60.3	-6.5	**68.2 ± 1.1**	67.8 ± 3.6	-0.4

Table 2: **Test ROC-AUC (%) performance of different GNN architectures with and without pre-training.** Without pre-training, the less expressive GNNs give slightly better performance

GNN Pre-Training: 异质图+对比式方法

Contrastive Pre-Training of GNNs on Heterogeneous Graphs, CIKM, 2021

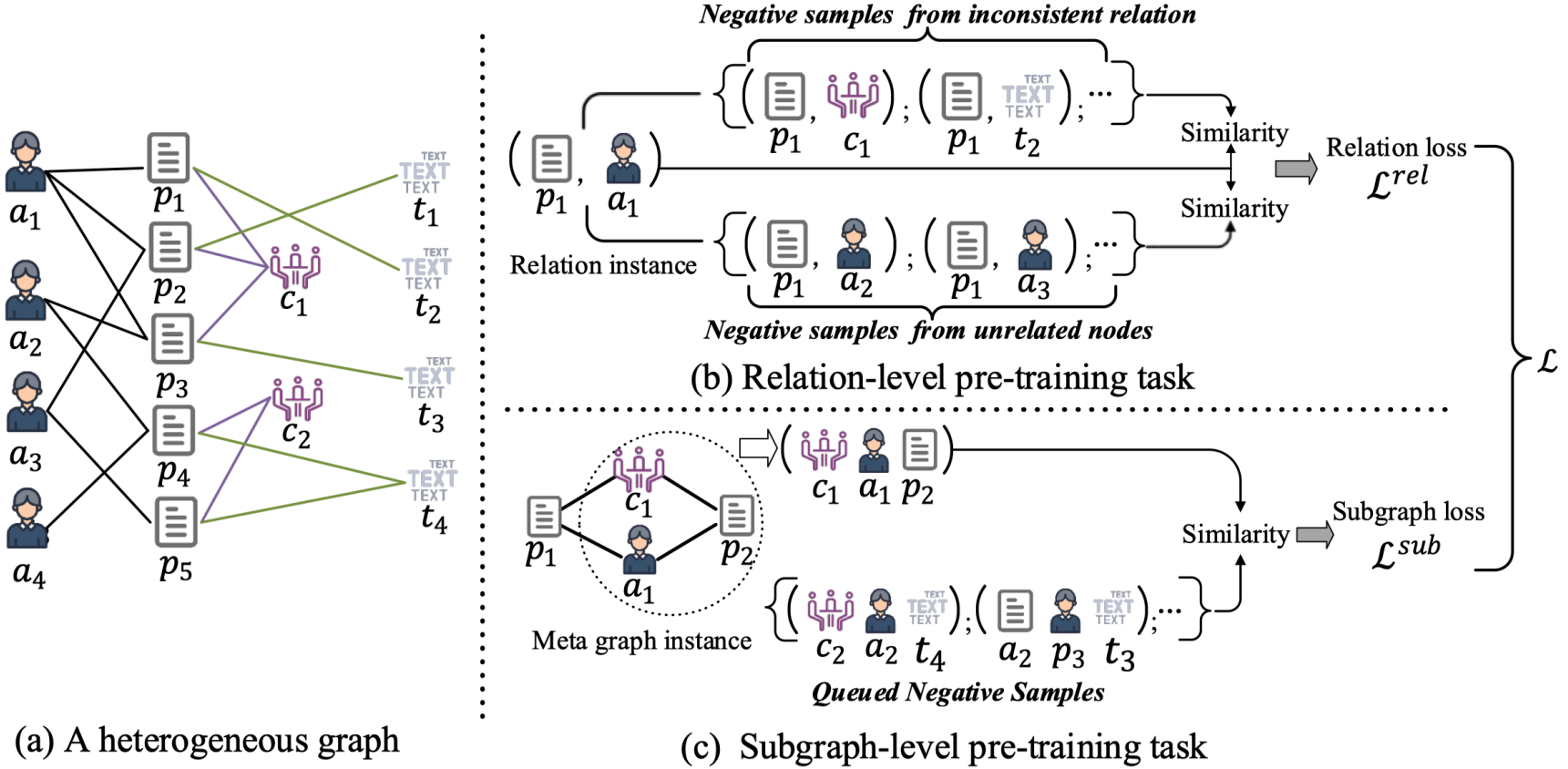
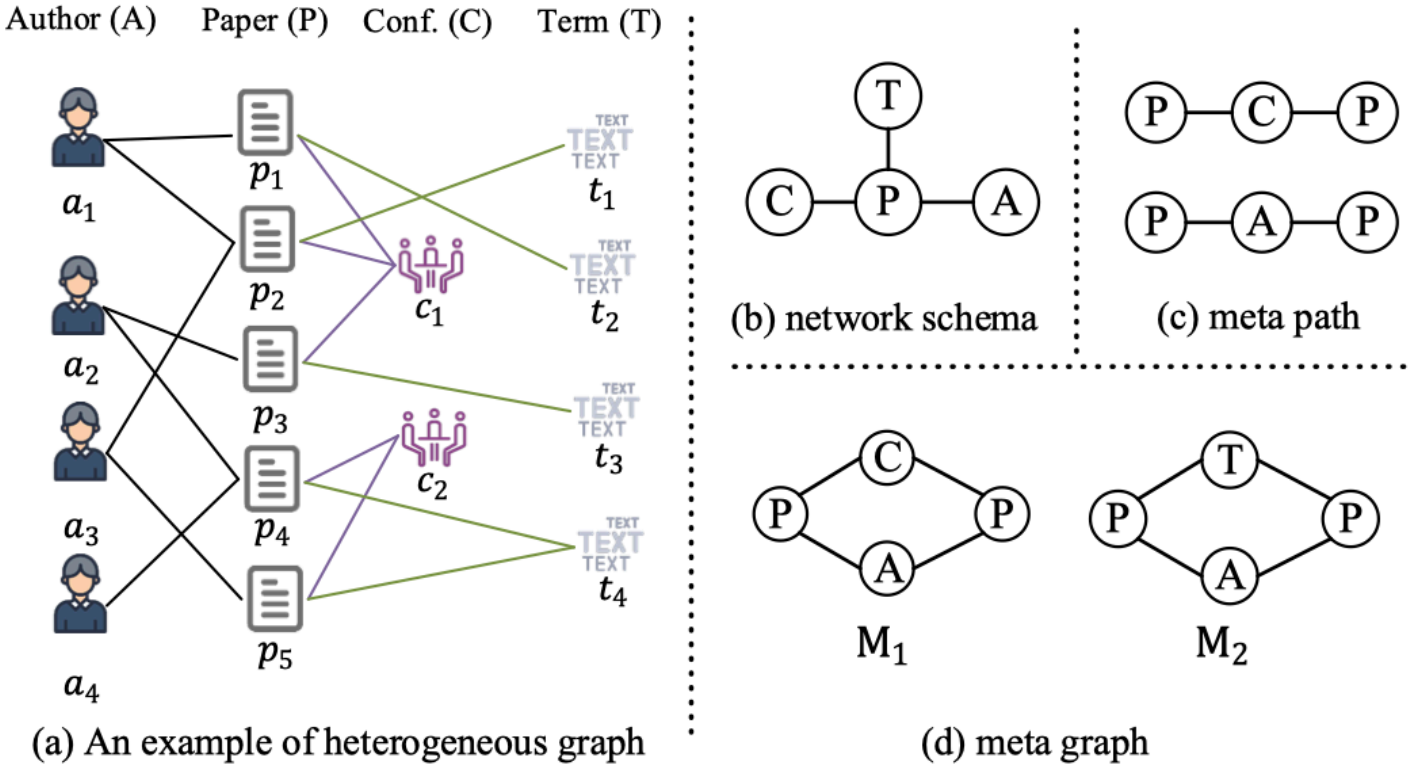


Figure 2: The overall framework of CPT-HG.

GNN Pre-Training: 异质图+对比式方法

Contrastive Pre-Training of GNNs on Heterogeneous Graphs, CIKM, 2021



采用meta graph 而不是 meta path的原因：

1. 元路径建模复杂语义和高阶结构的能力有限

2. 从一个节点出发，按照一个元路径可到达的节点太多，不够高效

GNN Pre-Training: 异质图+对比式方法

Contrastive Pre-Training of GNNs on Heterog CIKM, 2021

预训练任务：

1. relation-level

正样本：图中真实存在的边

$$\langle u, R, v \rangle \in \mathcal{P}^{rel}$$

负样本：

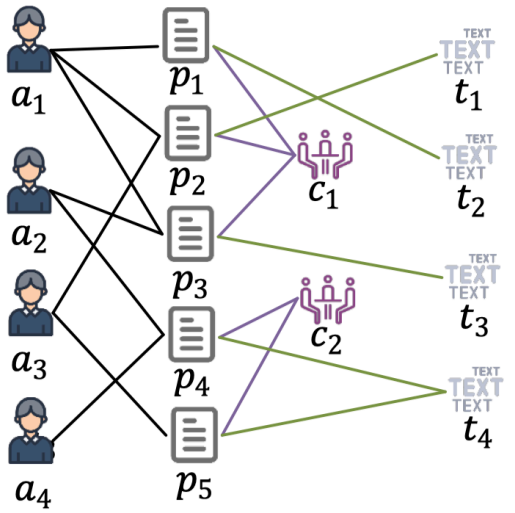
1. From inconsistent Relations

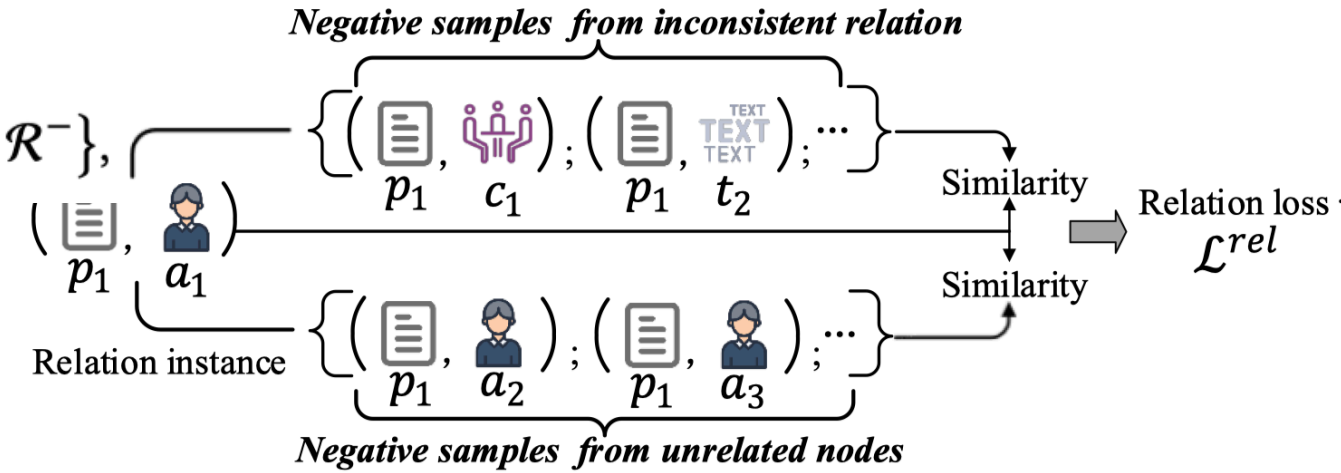
$$\mathcal{N}^{rel}_{\langle u,R,v \rangle} = \{\langle u, \varphi(u,w), w \rangle | (u,w) \in \mathcal{E}, \varphi(u,w) \in \mathcal{R}^{-}\},$$

2. From unrelated Nodes

$$\mathcal{N}^{node}_{\langle u,R,v \rangle} = \{\langle u, *, v^{-} \rangle | v^{-} \in \mathcal{V}\}.$$

where v^{-} is the the k-hop neighbors of node u





(b) Relation-level pre-training task

GNN Pre-Training: 异质图+对比式方法

Contrastive Pre-Training of GNNs on Heterogeneous Graphs, CIKM, 2021

预训练任务：

1. relation-level

正样本：图中真实存在的边

$\langle u, R, v \rangle \in \mathcal{P}^{rel}$

负样本：

1. From inconsistent Relations

$$\mathcal{N}^{rel}_{\langle u,R,v \rangle} = \{\langle u, \varphi(u,w), w \rangle | (u,w) \in \mathcal{E}, \varphi(u,w) \in \mathcal{R}^{-}\},$$

2. From unrelated Nodes

$$\mathcal{N}^{node}_{\langle u,R,v \rangle} = \{\langle u, *, v^{-} \rangle | v^{-} \in \mathcal{V}\}.$$

where v^{-} is the the k-hop neighbors of node u

优化目标：

$$\mathcal{L}^{rel1} = \sum_{\langle u,R,v \rangle \in \mathcal{P}^{rel}} -\log \frac{\exp\left(\mathbf{h}_u^{\top} \mathbf{W}_R \mathbf{h}_v\right)}{\sum_{i \in \{v\} \cup \{w | \langle u,R^{-},w \rangle \in \mathcal{N}^{rel}_{\langle u,R,v \rangle}\}} \exp\left(\mathbf{h}_u^{\top} \mathbf{W}_R \mathbf{h}_i\right)}$$

$$\mathcal{L}^{rel2} = \sum_{\langle u,R,v \rangle \in \mathcal{P}^{rel}} -\log \frac{\exp\left(\mathbf{h}_u^{\top} \mathbf{h}_v\right)}{\sum_{i \in \{v\} \cup \{v^{-} | \langle u,*,v^{-} \rangle \in \mathcal{N}^{node}_{\langle u,R,v \rangle}\}} \exp\left(\mathbf{h}_u^{\top} \mathbf{h}_i\right)}$$

$$\mathcal{L}^{rel} = \mathcal{L}^{rel1} + \mathcal{L}^{rel2}$$

GNN Pre-Training: 异质图+对比式方法

Contrastive Pre-Training of GNNs on Heterogeneous Graphs, CIKM, 2021

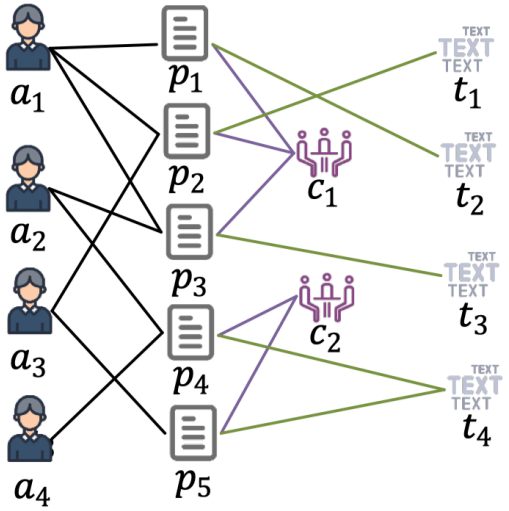
预训练任务：

2. subgraph-level

基于meta graph 构建子图

正样本：

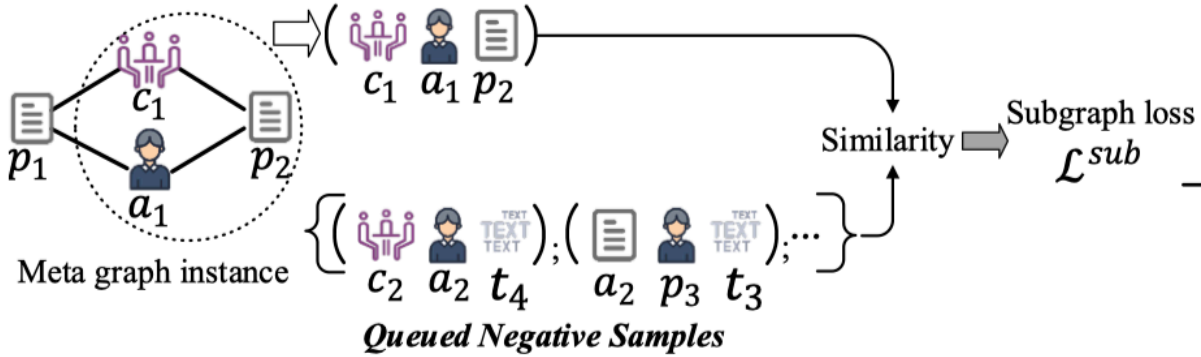
$$\mathcal{P}_u^{sub} = \bigcup_{m \in \mathcal{I}(M),\, M \in \mathcal{M}} m \setminus \{u\}$$



负样本：

$$\mathcal{N}^{sub} = [\mathcal{P}_i^{sub}(t-1), \mathcal{P}_j^{sub}(t-2), \cdots]$$

$$\mathcal{L}^{sub} = -\sum_{u \in \mathcal{V}} \sum_{P^+ \in \mathcal{P}_u^{sub}} \log \frac{\exp\left(\mathbf{h}_u^\top f(P^+)\right)}{\sum_{P \in \{P^+\} \cup \mathcal{N}^{sub}} \exp\left(\mathbf{h}_u^\top f(P)\right)},$$



GNN Pre-Training: 异质图+对比式方法

Contrastive Pre-Training of GNNs on Heterogeneous Graphs, CIKM, 2021

总优化目标：

$$\mathcal{L} = \mathcal{L}^{sub} + \lambda \mathcal{L}^{rel},$$

Adaptation to downstream tasks:

- Link Prediction: Fine-tuning
- Node Classification: Feature-based

GNN Pre-Training: 异质图+对比式方法

Contrastive Pre-Training of GNNs on Heterogeneous Graphs, CIKM, 2021

Dataset	Link Type	No pre-train	GAE	EgePred	DGI	GPT-GNN	CPT-HG	Improv.
DBLP	Paper-Term	12.34 ± 1.43	12.51 ± 0.71	12.61 ± 0.44	12.47 ± 0.68	12.71 ± 0.35	**13.06** ± 0.42	2.75%
Yelp	Business-Location	45.83 ± 0.42	45.92 ± 0.52	46.10 ± 0.31	45.57 ± 0.64	46.04 ± 0.75	**47.04** ± 0.71	2.03%
Aminer	Paper-Conference	39.23 ± 1.75	40.31 ± 0.78	39.86 ± 1.17	40.74 ± 1.35	41.37 ± 0.76	**42.17** ± 1.23	1.93%
	Paper-Author	5.63 ± 0.73	5.71 ± 0.41	5.62 ± 0.87	5.84 ± 0.52	6.02 ± 0.45	**6.43** ± 0.54	6.81%

Table 2: Experiment results (MRR(%) ± std) in link prediction task on the three datasets. The best method is bolded, and the second best is underlined.

Dataset	Labeled Node Type	No pre-train	GAE	EgePred	DGI	GPT-GNN	CPT-HG	Improve.
DBLP	Author	87.45 ± 0.43	90.56 ± 0.73	89.24 ± 0.57	88.26 ± 0.66	89.57 ± 0.45	**91.45** ± 0.54	0.98%
Aminer	Paper	92.17 ± 0.56	92.72 ± 0.32	93.41 ± 0.46	92.37 ± 0.25	93.75 ± 0.67	**96.32** ± 0.43	2.74%

Table 3: Experiment results (Accuracy(%) ± std) in the node classification task on DBLP and Aminer datasets. The best method is bolded, and the second best is underlined.

GNN Pre-Training: 异质图+对比式方法

Contrastive Pre-Training of GNNs on Heterogeneous Graphs, CIKM, 2021

Downstream Task	Link Prediction				Node Classification	
Dataset	DBLP	Yelp	Aminer		DBLP	Aminer
Link/Labeled Node Type	Paper-Term	Business-Location	Paper-Conference	Paper-Author	Paper	Author
No pre-train	12.34 ± 1.43	45.83 ± 0.42	39.23 ± 1.75	5.63 ± 0.73	87.45 ± 0.43	92.17 ± 0.56
CPT-HG$_{sub}$	12.65 ± 0.42	**47.15** ± 0.44	41.54 ± 0.33	6.04 ± 0.51	89.57 ± 0.61	94.14 ± 0.54
CPT-HG$_{rel}$	12.79 ± 0.56	46.74 ± 0.65	41.75 ± 0.65	6.24 ± 0.15	**92.45** ± 0.54	95.16 ± 0.32
CPT-HG	**13.06** ± 0.42	47.04 ± 0.71	**42.17** ± 1.23	**6.43** ± 0.54	91.45 ± 0.54	**96.32** ± 0.43

Table 4: Analysis of different ablated models in various downstream tasks.

GNN Pre-Training: 异质图+对比式方法

Contrastive Pre-Training of GNNs on Heterogeneous Graphs, CIKM, 2021

Base Model	No pre-train	CPT-HG	Improvement
HGT	5.63 ± 0.73	6.42 ± 0.54	14.0%
RGCN	4.15 ± 0.43	4.49 ± 0.50	7.55%
GCN	4.79 ± 0.81	5.14 ± 0.62	7.31%
GAT	4.83 ± 0.77	4.32 ± 0.89	-10.6%
GraphSAGE	5.47 ± 0.52	6.02 ± 0.62	10.24%

Table 5: Analysis of different GNN architectures in link prediction on Aminer dataset.

GNN Pre-Training: 异质图+对比式方法

Contrastive Pre-Training of GNNs on Heterogeneous Graphs, CIKM, 2021

Model	Accuracy	Improvement
No pre-train	92.17 ± 0.56	-
CPT-HG$_{PAP}$	91.52 ± 0.56	-0.7%
CPT-HG$_{PCP}$	90.45 ± 0.41	-1.9%
CPT-HG$_{PATP}$	93.12 ± 0.52	+1.0%
CPT-HG$_{PACP}$	94.71 ± 0.46	+2.6%
CPT-HG	96.32 ± 0.56	+4.5%

Table 6: Analysis of different meta graphs in node classification on Aminer dataset.

Percentage	MRR	Improvement
No pre-train	5.63 ± 0.56	-
10%	5.54 ± 0.16	-1.5%
50%	5.87 ± 0.41	+4.2%
100%	6.43 ± 0.56	+14.0%

Table 7: Compare the pre-training performance Gain with different percentage of pre-training datasets. Evaluate the Paper-Author link prediction on Aminer.

GNN Pre-Training: 异质图+对比式方法

Pre-training on Large-Scale Heterogeneous Graph, KDD, 2021

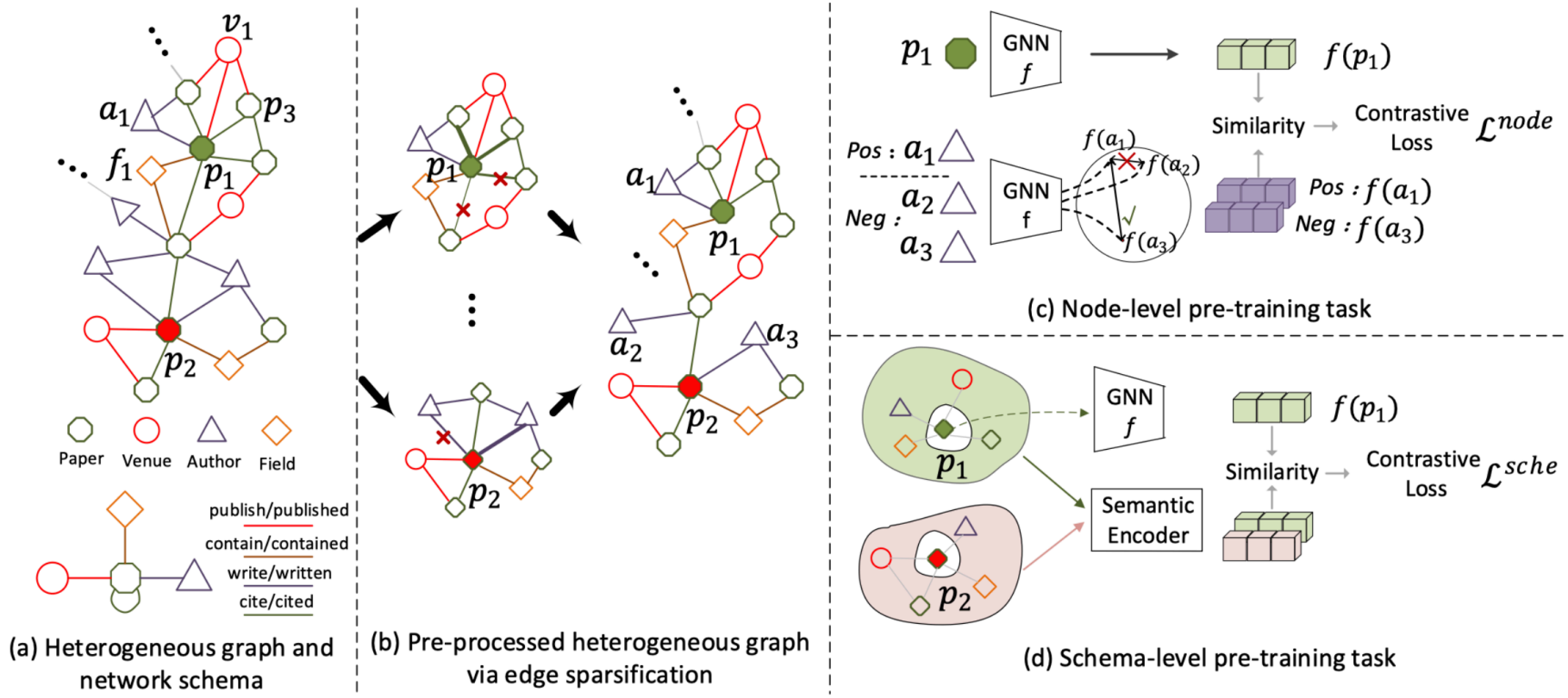


Figure 1: The overall framework of PT-HGNN.

GNN Pre-Training: 异质图+对比式方法

Pre-training on Large-Scale Heterogeneous Graph, KDD, 2021

边级别：

$$\mathcal{N}^{node}_{\langle u,R,v\rangle} = \{\langle u, R, v^-\rangle \mid \phi(v) = \phi(v^-), (u, v^-) \notin \mathcal{E}, Sim(v, v^-) \leq \delta\},$$

$$\mathcal{L}^{node}_{u,R} = -\log \frac{\exp\left(\mathbf{h}_u^\top \mathbf{W}_R \mathbf{h}_v/\tau\right)}{\sum_{i\in\{v\}\cup\{w|\langle u,R,w\rangle\in\mathcal{N}^{node}_{\langle u,R,v\rangle}\}} \exp\left(\mathbf{h}_u^\top \mathbf{W}_R \mathbf{h}_i/\tau\right)},$$

子图级别：

$$\mathcal{P}^{sche}_u = \bigcup_{\mathbf{s}\in I(u)} \mathbf{s}\backslash\{u\},$$

$$\mathcal{N}^1_u = \{\mathcal{P}^{sche}_{u^-} \mid u^- \in \mathcal{V}_B, u \neq u^-, \phi(u) = \phi(u^-)\}.$$

$$\mathcal{N}^2_u = \{\mathcal{P}^{sche}_v \mid \phi(u) = \phi(v), v \in \mathcal{V}^{t-1}_B\},$$

$$\mathcal{N}^{sche}_u = \mathcal{N}^1_u \cup \mathcal{N}^2_u.$$

$$\mathcal{L}^{sche}_u = \sum_{\mathbf{s}^+\in\mathcal{P}^{sche}_u} \log \frac{\exp\left(\mathbf{h}_u^\top \mathbf{c}^{\mathbf{s}^+}/\tau\right)}{\sum_{\mathbf{s}\in\{\mathbf{s}^+\}\cup\mathcal{N}^{sche}_u} \exp\left(\mathbf{h}_u^\top \mathbf{c}^{\mathbf{s}}/\tau\right)},$$

GNN Pre-Training: 异质图+对比式方法

Pre-training on Large-Scale Heterogeneous Graph, KDD, 2021

Sparsification of Large-Scale Heterogeneous Graph

$$\Pi^{R} = \alpha \mathbf{I} + (1-\alpha) S \Pi^{R^{-1}}$$

$$\Pi^{R^{-1}} = \beta \mathbf{I} + (1-\beta) S^{T} \Pi^{R}$$

$$\tilde{A}^{R}_{uj} = \begin{cases} 1, & \text{if } \Pi^{R}_{uj} > 0 \text{ and } (u, j) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases} .$$

GNN Pre-Training: 异质图+对比式方法

Self-supervised Heterogeneous Graph Neural Network with Co-contrastive Learning, KDD, 2021

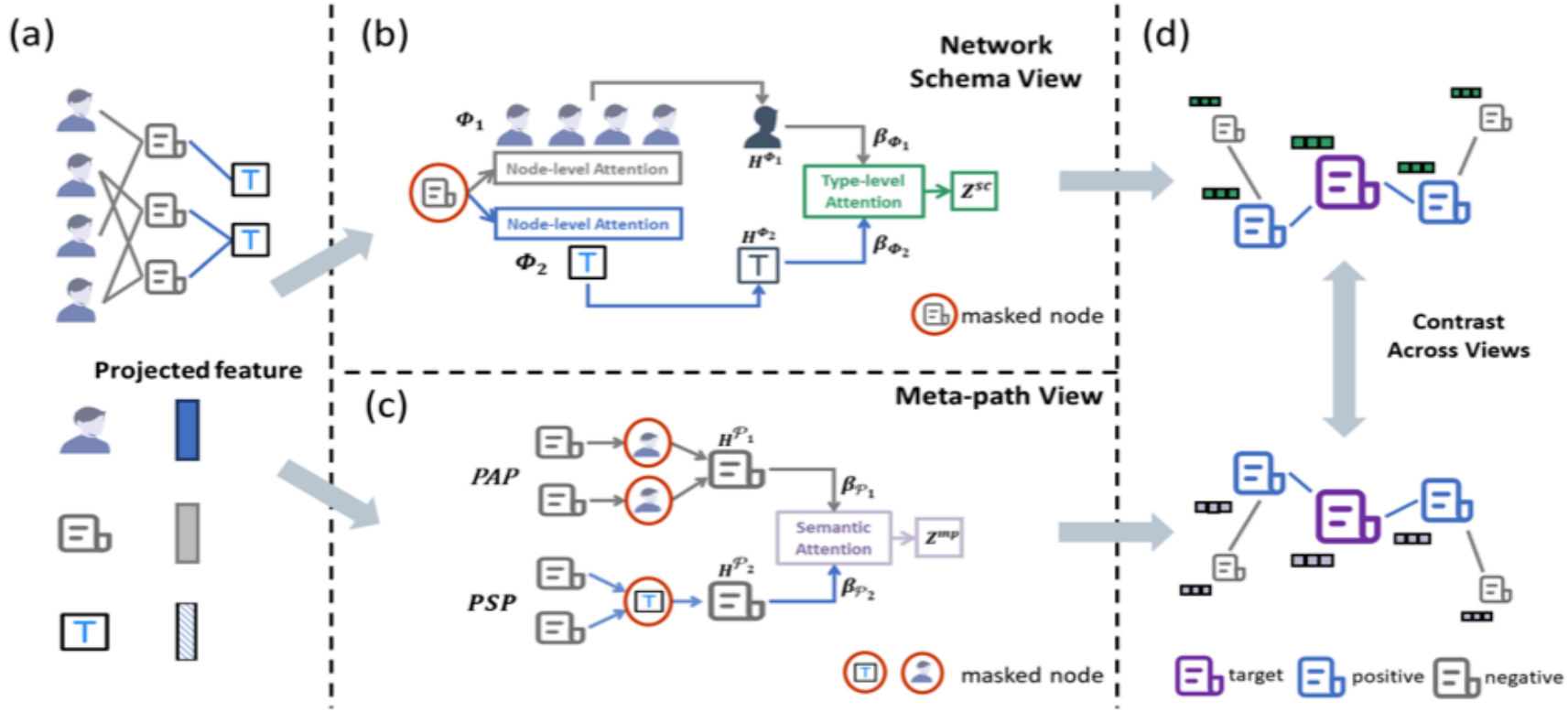


Figure 2: The overall architecture of our proposed HeCo.

GNN Pre-Training

- pretext任务
  - 自监督范式：生成式/对比式
  - 学习的对象：attribute/structure
  - 学习的层次：local/context/global (node/edge/subgraph/graph)
- 适配下游任务的方式：feature-based/fine-tuning
- 图的类型：同质图/异质图